

WHAT IS CLAIMED:

1. A method of emulating a network of two or more distinct types of logic systems comprising:

providing a plurality of actual logic systems of at least two distinct types;

providing a communication channel to said actual logic systems; and

running logic instructions on said actual logic systems whereby two or more said actual logic systems respond on said communication channel as though each were multiple logic systems, wherein an actual logic system responds as though it were multiple logic systems similar to its type.

2. The method according to claim 1 further comprising:

at at least one of said actual logic systems, responding to multiple incoming addresses on said communication channel as though said at least one logic system were multiple logic systems.

3. The method according to claim 1 further comprising:

on at least one actual logic system, providing varying responses.

4. The method according to claim 3 wherein said responses vary based on an incoming address.

5. The method according to claim 3 wherein said varying responses comprise varying time and use characteristics.

6. The method according to claim 1 wherein said responses of said two or more actual logic systems are altered over time to emulate characteristics of real networks.

7. The method according to claim 1 wherein said emulation is used to deceive unauthorized users trying to access one or more protected logic systems.

8. The method according to claim 7 wherein said emulation is used to deceive unauthorized users trying to access one or more protected logic systems by providing deceptive responses to unauthorized datagrams so as to lead an unauthorized user to believe the user has accessed an actual computer system.

9. The method according to claim 1 wherein said emulation is controllable from one or more control systems.

10. The method according to claim 9 wherein said one or more control systems comprise one or more distributed control systems.

11. The method according to claim 1 wherein said distinct types comprise different operating systems.

12. The method according to claim 1 wherein said distinct types comprise:

different operating systems; and

different hardware platforms.

13. A computer network emulation system comprising:

two or more emulation computer systems of at least two distinct types;

a network able to deliver datagrams to said two or more emulation computer systems; and

wherein two or more of said two or more emulation computer systems provides emulation responses of multiple emulated computer systems at multiple addresses, each emulation computer system providing emulation responses of emulated computers appropriate to said emulation computer system's type.

14. The system according to claim 13 further comprising:

a transmission control system able to establish and vary delivery paths to said emulation computer systems.

15. The system according to claim 13 further comprising:

an emulation control system able to establish and vary emulations provided by said two or more emulation computer systems.

16. A method of connecting an emulation subnetwork to a network through an emulation wall comprising:

receiving a datagram at an outside of an emulation wall;

determining that said datagram should be handled in said emulation subnetwork;

translating an original address indication of said datagram into a proxy address indication;

passing said datagram into said emulation subnetwork while translating said proxy address into an emulation original address indication; and

routing said packet in said emulation subnetwork based on said emulation original address indication.

17. The method according to claim 16 further comprising:

receiving a response at an inside of said emulation wall from said emulation subnetwork;

translating a response emulation original address indication to a response proxy address indication; and

passing said datagram into said network from said emulation subnetwork while translating said response proxy address indication back to a response original address indication.

18. The method according to claim 16 wherein address indications comprise source and destination addresses according to a standard networking protocol.

19. The method according to claim 18 wherein address indications comprise source and destination addresses according to an IP networking protocol.

20. The method according to claim 16 wherein said emulation original address indication is identical to said original address indication.

21. The method according to claim 16 wherein said passing comprises:

receiving said datagram at a first standard network gateway;

translating said original address indication of said datagram into a proxy address indication at said first standard network gateway using a standard translation procedure;

routing said datagram with said proxy address indication to a second standard network gateway;

translating said proxy address indication of said datagram into said emulation original address indication at said second standard network gateway using a standard translation procedure;

and

forwarding said datagram with said emulation original address indication to said emulation subnetwork.

22. An emulation wall connecting an emulation subnetwork to an outside network comprising:

an outside layer able to detect datagrams in said outside network to be handled in said emulation subnetwork;

a transport module, able to transport said datagrams into an internal emulation network while preserving their original source and destination address;

one or more emulation systems on said emulation subnetwork able to receive said datagrams with original source and destination address and to generate appropriate response datagrams onto said emulation subnetwork; and

an inside layer able to detect response datagrams to be transferred to said outside network; said transport facility able to transport said datagrams to said outside network.

23. The device according to claim 22 wherein said outside layer and said inside layer are two network interfaces in a specialized network translation device and said transport module is implemented in specialized datagram handling logic in said device:

24. The device according to claim 22 wherein:

said outside layer comprises an outside interface of a first standard network address translation module;

and said inside layer comprises an inside interface of a second standard network address translation module;

said transport module is implemented by performing a first translation within said first standard network address translation module into a proxy address, communicating a translated datagram to said second standard network address translation module and translating from the proxy address back to the original address on said emulation subnetwork.

25. A method for countering attacks in a network comprising:

at an emulation computer system in said network, accepting network protocol datagrams that are unauthorized; and

responding, by said computer system, to unauthorized datagrams using different emulations so that an attacker perceives that a number of different computer systems within said network have been reached.

26. The method according to claim 25 wherein a network protocol datagram is detected as unauthorized by detecting that said datagram is addressed to a computer system that does not actually exist on said network.

27. The method according to claim 25 further comprising:

at a normal computer in said network, detecting unauthorized network protocol datagrams addressed to said normal computer; and

routing said unauthorized network protocol datagrams addressed to said normal computer to said emulation computer using an address translation, for response by said emulation computer.

28. The method according to claim 25 wherein an apparent emulated architecture changes with time or incoming viewpoint just as a large scale computer network changes with time and viewpoint.

29. A method of providing deception at a computer system on a network comprising:

accepting, at said computer system, network protocol datagrams addressed to different computers; and

responding, by said computer system, to received datagrams using different deception emulations so that a receiver perceives that a number of different computer systems have been reached.

30. A method of protecting a computer network against unwanted attacks comprising:
 routing datagrams addressed to non-existing computers to a deception system; and
 at said deception system, responding to said datagrams using varying emulations.

31. The method according to claim 30 wherein said emulations vary based on one or more
 parameters including datagram addresses, time, or usage statistics.

32. The method according to claim 30 wherein an emulation at a particular IP address
 translates the same services differently for different remote access points, creating for some, the
 illusion of a first network, for others the illusion of a service system with a vulnerable deception
 target, and for still others, access to other systems.

33. A method enabling multiple processor processing across physically distributed computer
 systems comprising:

at a first gateway between a first multiple processor network and an intermediate network,
 receiving from said first multiple processor network a multiple processor-addressed
 datagram to computers not on said first network;

at said first gateway, translating a multiple processor address indication into an intermediate
 address indication;

transmitting said datagram with an intermediate address indication over an intermediate
 network;

at a second gateway between said intermediate network and a second multiple processor
 network, receiving said datagram with an intermediate address indication;

at said second gateway, translating an intermediate address indication back to a multiple
 processor address indication; and

transmitting said datagram with a multiple processor address indication on said second
 multiple processor network so that said datagram can be received on a computer on said
 second multiple processor network.

34. The method according to claim 33 wherein said first gateway translates addresses into a
 form that uses the port number in an IP protocol to hold the last two elements of the IP address of
 the remote computer within its subnetwork and has a lookup table to indicate which remote IP
 address to use for each remote subnetwork.

35. The method according to claim 33 wherein said intermediate network comprises the
 world-wide Internet.

36. The method according to claim 33 wherein said first gateway and said second gateway
 are standard network gateway devices able to perform a standard gateway address translation.

37. The method according to claim 33 wherein said multiple computer processing comprises parallel multiple instruction multiple data (MIMD) processing.

38. The method according to claim 33 wherein said first multiple processor network and said second multiple processor network each require assignment of only one address in said intermediate network to enable multiple process processing.

39. The method according to claim 38 wherein said first gateway and said second gateway each translate data packets to up to approximately 64,000 computer systems.

40. The method according to claim 38 wherein address translations can be altered at said gateways to reconfigure distributed computing transparently to any computer systems participating in said distributed computing.

41. A distributed computer processing system comprising:
a plurality of computer system groups, each group comprising at least one computer participating in said distributed computer processing system, with computers in each group reachable by a group local address;
a plurality of interfaces to an intermediate network that can transport data between said plurality of groups; and
one or more address translation modules between each of one or more of said groups and said intermediate network; said address translation modules at one end able to translate a first local address from a first group computer to a second group computer to an appropriate address to reach said second group via said intermediate network and at said second group translating said intermediate address back to a local network address.

42. The system according to claim 41 further comprising:
a first group with a group of first local addresses;
a first address translation module with an interface to communicate datagrams with said first group and an interface to communicate datagrams to an intermediate network;
a second group with a group of second local addresses;
a second address translation module with an interface to communicate datagrams with said second group and an interface to communicate datagrams to an intermediate network;
wherein first local addresses and second local addresses can be translated into a common addressing scheme with no overlapping assignment of addresses.

43. The system according to claim 41 further comprising:

wherein a computer in a local group can communicate using its local groups addressing scheme not only with other computers in its local group but also with computers in other local groups; and

wherein an address translation module in a local group detects datagrams with local group addressing indicating computers in other groups and translates the addressing of those datagrams and places them on a network that can reach a second address translation module, said second translation module translating to a second local group.

44. The system according to claim 41 further wherein said translation is transparent to said first computer so that said first computer perceives it is part of a local group encompassing all distributed computer systems.

45. The system according to claim 41 further wherein said datagram is detected by said first address translation module and transmitted on said intermediate network with a translated address to an appropriate second address translation module.

46. The system according to claim 41 further wherein said second address translation module receives said datagram and translates its address to an address in said second group and transmits said address to said second group.

47. The system according to claim 41 further wherein a computer in a local group perceives that it can communicate with a large number of computers using a local addressing scheme, but where two or more of said locally addressed computers have addresses translated by said gateway.

48. A method enabling multiple processor computing comprising:
 deploying a plurality of MIMD logic processing modules to a plurality of computer systems, said modules aware of a local MIMD addressing scheme for reaching other MIMD logic processing modules;

wherein said plurality of computer systems are deployed in a plurality of local groups;

deploying a plurality of address translation modules between said local groups and an intermediate communication system;

wherein an address translation module is able to detect an MIMD packet at a local group that it addressed to an MIMD logic processing module in a different local group;

wherein said address translation module translates said detected packet to an intermediate address deliverable to a different address translation module over said intermediate communication system; and

wherein said different address translation module receives a packet over said intermediate transmission system and translates it to a local MIMD addressing scheme at a different local group.

49. The method according to claim 48 wherein said local MIMD addressing scheme and said intermediate address are both in a standard IP addressing protocol.

50. The method according to claim 48 wherein said MIMD logic processing modules communicate packets unaware of whether other MIMD logic processing modules are local or reached through address translations.

51. The method according to claim 48 wherein said plurality of address translation modules are responsible for keeping track of to which local groups MIMD logic processing modules belong.

52. The method according to claim 49 wherein said MIMD address scheme comprises a plurality of class B subnetwork addresses of the form 10.#.*.*, where 10 indicates any dedicated class A subnetwork, and # indicates one or more class B subnetworks.

53. The method according to claim 49 wherein said intermediate address scheme comprises a plurality of individual assigned IP addresses of the form a.b.c.d and wherein said address translation modules manage a mapping between each assigned intermediate address and an MIMD 10.#.*.* network.

54. The method according to claim 48 wherein said address translation modules are standard address translation gateway devices.

55. The method according to claim 48 wherein said intermediate communication system comprises the Internet.

56. The method according to claim 48 wherein multiple address translation makes the physical location of distributed computing resources transparent and automatic to the programmer of distributed system software.

57. The method according to claim 48 wherein the method eliminates the need for an MIMD logic processor or programmer to differentiate between local and distant resources or to know anything about the topology of the network.

58. The method according to claim 48 wherein assignments of MIMD processing modules or address translation modules can vary over time to compensate for failures without the need for other MIMD processing modules to be aware of the variations.

59. The method according to claim 48 wherein the emulation component translates addresses into a form that uses the port number in the IP protocol to hold the last two elements of the IP

address of the remote computer within its subnetwork and has a lookup table to indicate which remote IP address to use for each remote subnetwork.

60. A method of obscuring a path of a datagram through a network comprising:

initially directing a datagram to a first translation module in said network;

at said first translation module, translating said datagram's addressing directing said datagram towards a final translation module in said network; and

at said final translation module, translating said datagram's addressing directing said datagram to a final destination.

61. The method according to claim 60 further comprising:

at said first translation module, translating said datagram's addressing towards an intermediate translation module in said network; and

at said intermediate translation module, translating said datagram's addressing directing said datagram towards a final destination.

62. The method according to claim 60 wherein at each translation module, translations are stored internally and are not accessible outside of said translation module.

63. The method according to claim 60 wherein each translation module comprises a standard address translation gateway.

64. A method for providing deception in a computer network comprising:

passing a datagram received with a non-legitimate characteristic (such as address, port, or improper characteristics) into a deception network.

65. The method according to claim 64 further comprising:

if a controlling system of said deception network has marked said datagram type as something to be ignored, dropping said datagram;

if possible handling said datagram at an initial interface so that said datagram never reaches a deception system; and

otherwise, if said datagram is to be passed into the more detailed deception, translating said datagram by way of a proxy into internal address indications.

66. A method enabling accessing distributed computing resources residing in a plurality of different locations to form a virtual local area network environment comprising:

at a first translation point between a first group location and an intermediate network, receiving from a computing device at said first location a datagram with a first local address indicating a destination not at said first group location;

at said first translation point, translating said local address indication of said datagram into an intermediate address indication;
 transmitting said datagram with an intermediate address indication over an intermediate network to a second translation point;
 5 at said second translation point, translating said intermediate address of said datagram to a second local address; and
 using said local address in a local addressing scheme to reach a desired computer system in said second group.

67. The method according to claim 66 wherein said first local address, said intermediate
 10 address, and said second local address are IP addresses.

68. The method according to claim 66 wherein said intermediate network comprises the world-wide Internet.

69. The method according to claim 66 wherein said first translation point and said second translation point are standard network gateway modules able to perform a standard gateway
 15 address translation.

70. The method according to claim 66 wherein it is intended to aide a user by abstracting the physical location of distributed computers.

71. The method according to claim 66 wherein internal users are deceived into an abstraction that there is no intervening intermediate network infrastructure, while external users can clearly
 20 see that the local network is fragmented over many locations.

72. / A distributed computing system comprising:
 a plurality of address translation modules, each address translation module effectively residing
 between one of a plurality of local communication groups and an intermediate communication infrastructure;
 25 a plurality of address detecting modules, each able to detect datagrams at a local communication group indicating devices not in that local group;
 wherein a first of said address translation modules receives a detected datagram and translates a local address to an intermediate address and transfers a translated datagram on an intermediate communication infrastructure to a second of said address translation modules;
 30 and
 wherein said second of said address translation modules receives a datagram from said intermediate communication infrastructure and translates and intermediate address to a second local addressing scheme.

73. The system according to claim 72 wherein said translation modules include translation logic for translating detected local addresses to an appropriate intermediate address to reach a desired computing device.